

# 대한민국 특허청

## KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원 번호 : 10-2003-0004366  
Application Number

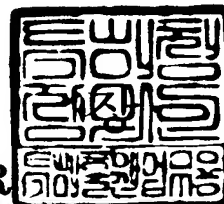
출원 년 월 일 : 2003년 01월 22일  
Date of Application JAN 22, 2003

출원인 : (주)넥슨  
Applicant(s) NEXON CO., LTD.

2003      년      05      월      21      일

특      허      청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2003.01.22
【발명의 명칭】	사용자 응용 프로그램의 제어 방법
【발명의 영문명칭】	METHOD FOR CONTROLLING USER APPLICATION PROGRAM
【출원인】	
【명칭】	( 주)넥슨
【출원인코드】	1-1998-611360-3
【대리인】	
【성명】	정태영
【대리인코드】	9-2001-000339-7
【포괄위임등록번호】	2002-075778-5
【발명자】	
【성명의 국문표기】	정상원
【성명의 영문표기】	CHUNG, Sang won
【주민등록번호】	700313-1024814
【우편번호】	143-210
【주소】	서울특별시 광진구 광장동 현대파크빌 1003동 2302호
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 정태영 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	15 면 15,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	18 항 685,000 원
【합계】	729,000 원
【감면사유】	중소기업
【감면후 수수료】	364,500 원

1020030004366

출력 일자: 2003/5/22

【첨부서류】

1. 중소기업기본법시행령 제2조에 의한 중소기업에 해당함을 증명하는 서류\_1통

**【요약서】****【요약】**

본 발명은 사용자 단말기에서 수행되는 사용자 응용 프로그램의 제어 방법에 관한 것으로, 더욱 상세하게는 사용자 응용 프로그램의 수행 중 필요한 데이터를 소정의 서버로부터 다운로드 하면서 사용자 응용 프로그램을 수행할 수 있도록 하는 사용자 응용 프로그램의 제어 방법에 관한 것이다.

클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램을 제어하기 위한 본 발명은, 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 후킹(hooking)하는 단계, 상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있는지 판단하는 단계, 상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있으면, 상기 파일의 데이터를 상기 사용자 응용 프로그램에게 전달하는 단계, 및 상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있지 않으면, 상기 파일이 저장되어 있는 소정의 서버-상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결됨-로부터 상기 파일의 데이터를 수신하여 상기 사용자 응용 프로그램에 전달하는 단계를 포함한다.

본 발명에 따르면, 종래의 패키지 형태의 소프트웨어 제품을 간편하게 온라인 소프트웨어화하여 온라인 스트리밍(on-line streaming)의 효과를 가져오고, 네트워크를 통하여 서버와의 연결을 유지함으로써 외부로부터의 해킹 및 크래킹을 효과적으로 방지하여 불법적인 소프트웨어의 사용을 방지할 수 있다.

**【대표도】**

도 3

**【색인어】**

온라인 스트리밍(online streaming), 온라인 인증, 온라인 스트리밍 파일 시스템

**【명세서】****【발명의 명칭】**

사용자 응용 프로그램의 제어 방법{METHOD FOR CONTROLLING USER APPLICATION PROGRAM}

**【도면의 간단한 설명】**

도 1은 본 발명에 따라 클라이언트 컴퓨터와 파일 서버의 네트워크 연결을 도시한 도면.

도 2는 본 발명에 따라 클라이언트 컴퓨터에 포함된 사용자 응용 프로그램, 온라인 스트리밍 파일 라이브러리 및 네트워크 인터페이스의 관계를 도시한 블록도.

도 3은 본 발명에 따라 클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램으로 부터의 파일 판독 요청을 처리하는 순서를 도시한 순서도.

도 4는 본 발명에 따라 클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램으로 부터의 파일 기록 요청을 처리하는 순서를 도시한 순서도.

도 5는 본 발명에 따라 사용자 응용 프로그램이 접근하는 파일 내 데이터 블록을 프로파일링(profiling)하여 우선 파일(priority file)을 구성하는 경우, 데이터 블록과 우선 파일의 관계를 도시한 도면.

도 6은 본 발명의 일 실시예에 따라 구성된 인덱스 테이블을 도시한 도면.

<도면의 주요 부분에 대한 부호의 설명>

102, 103: 클라이언트 컴퓨터

104: 서버

201: 사용자 응용 프로그램

202: 온라인 스트리밍 파일 라이브러리

203: 네트워크 인터페이스

501: 데이터 파일

502, 503, 504: 우선 파일

**【발명의 상세한 설명】**

**【발명의 목적】**

**【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<15>        본 발명은 사용자 단말기에서 수행되는 사용자 응용 프로그램의 제어 방법에 관한 것으로, 더욱 상세하게는 사용자 응용 프로그램의 수행 중 필요한 데이터를 소정의 서버로부터 다운로드 하면서 사용자 응용 프로그램을 수행할 수 있도록 하는 사용자 응용 프로그램의 제어 방법에 관한 것이다.

<16>        종래 기술에서 패키지 게임(오프라인 게임)은 게임 프로그램 및 상기 게임에 필요한 데이터를 모두 CD 등의 저장 매체에 저장하여 사용자에게 배포된다. 그런데, 이러한 패키지 게임을 포함한 패키지 형태의 소프트웨어 제품은 불법 복제되어 정당한 권원 없는 사용자에게 의하여 사용되는 경우가 많다. 따라서 이러한 패키지 형태의 소프트웨어 제품을 온라인화하면 이러한 불법 복제를 막을 수 있을 뿐만 아니라, 자동 업데이트와 같은 여러 가지 온라인 소프트웨어의 장점을 이용할 수 있다. 이러한 이유로 이러한 패키지 형태의 소프트웨어 제품을 간편하게 온라인 소프트웨어로 전환할 수 있도록 하는 방법이 요구되고 있다.

<17> 또한 온라인 게임(on-line game)의 경우에도 상기 온라인 게임의 수행에 필요한 모든 파일을 사용자 컴퓨터에 다운로드 받은 후에야 상기 온라인 게임을 수행할 수 있다. 그러나 온라인 게임이 더욱 발전되고 복잡해질수록 사용자가 초기에 다운로드 해야 할 데이터의 양은 더욱 더 증가하고 있다. 최근에는 온라인 게임을 수행하기 위하여 초기에 다운로드 해야 하는 데이터의 양이 200 Mbyte를 넘는 경우도 빈번히 발생하고 있다. 이와 같이 초기에 다운로드 해야 하는 데이터의 양이 너무 크기 때문에, 온라인 게임의 최초 사용자가 쉽게 온라인 게임에 참여하지 못하게 되는 단점이 있다. 따라서 전체 온라인 게임을 수행하기 위하여 필요한 데이터의 크기가 크더라도, 그 중 일부의 적은 데이터만을 다운로드 한 후 온라인 게임을 수행할 수 있는 방법이 요구되고 있다.

**【발명이 이루고자 하는 기술적 과제】**

<18> 본 발명은 상기와 같은 종래 기술의 문제점을 해결하기 위하여 안출된 것으로서, 종래의 패키지 형태의 소프트웨어 제품을 간편하게 온라인 소프트웨어화하여 온라인 스트리밍(on-line streaming)의 효과를 가져오고, 네트워크를 통하여 서버와 연결을 유지함으로써 외부로부터의 해킹 및 크래킹을 효과적으로 방지하여 불법적인 소프트웨어의 사용을 방지하는 것을 목적으로 한다.

<19> 또한, 본 발명은 종래의 패키지 소프트웨어를 간단히 온라인화함으로써, 주기적으로 상기 패키지 소프트웨어의 불법 복제 여부를 감시할 수 있고, 만일 불법 복제 제품인 경우에는 상기 패키지 소프트웨어의 동작을 정지시킴으로써, 불법 복제를 막는 것을 목적으로 한다. 또한, 본 발명의 목적은 패키지 소프트웨어를 온라인화함으로써 패키지 소프트웨어의 과금을 사용 시간 당 과금이 가능하도록 하는 것이다.



<20> 또한, 본 발명의 목적은 온라인 소프트웨어에서 파일 전체가 아닌 파일의 블록 단위로 다운로드 하여 초기에 다운로드 하여야 하는 데이터의 크기를 최소화시킴으로써, 사용자들이 초기에 쉽게 게임을 즐길 수 있도록 하는 것이다.

<21> 또한 본 발명의 목적은 온라인 소프트웨어가 요구하는 데이터의 패턴을 조사하여, 이에 따라 다운로드 되는 데이터를 최적화하고, 사용자 응용 프로그램이 데이터의 판독을 요청하지 않는 경우에도, 상기 조사된 데이터의 패턴에 따라 미리 백그라운드 피투피(Background P2P)를 통하여 필요한 데이터를 미리 다운로드 받음으로써 사용자가 데이터의 다운로드를 위하여 기다리는 시간을 최소화하고, 또한 서버의 대역폭(bandwidth) 사용을 최소화하는 것이다.

#### 【발명의 구성 및 작용】

<22> 상기의 목적을 이루고 종래기술의 문제점을 해결하기 위하여, 클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램을 제어하기 위한 본 발명은, 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하는 단계, 상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있는지 판단하는 단계, 상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있으면, 상기 파일의 데이터를 상기 사용자 응용 프로그램에게 전달하는 단계, 및 상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있지 않으면, 상기 파일이 저장되어 있는 소정의 서버-상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결됨-로부터 상기 파일의 데이터를 수신하여 상기 사용자 응용 프로그램에 전달하는 단계를 포함한다. 본 발명의 일 실시예에 따르면, 상기 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하는 단계는, 상기 사용자 응용 프로그램으로부터의 상

기 파일에 대한 상기 파일 판독 요청을 후킹(hooking)하거나 상기 파일 판독 요청을 처리하는 원래의 함수를 다른 함수로 매핑하는 단계를 포함한다.

<23>       본 발명의 일 측면에 따르는 사용자 응용 프로그램 제어 방법은 사용자 응용 프로그램으로부터의 파일에 대한 파일 기록 요청을 수신하는 단계, 상기 파일이 소정의 서버-상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결됨-에 업로드(upload)될 필요가 있는지 판단하는 단계, 상기 파일이 상기 소정의 서버에 업로드될 필요가 있으면, 상기 파일을 상기 소정의 서버로 업로드하는 단계, 및 상기 파일이 상기 소정의 서버에 업로드될 필요가 없으면, 상기 클라이언트 컴퓨터에 상기 파일을 기록하는 단계를 포함하는 사용자 응용 프로그램 제어 방법을 제공한다.

<24>       본 발명의 일 측면에 따르면, 상기 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하는 단계, 상기 파일 판독 요청에 응답하여 클라이언트 컴퓨터에 저장된 상기 사용자 응용 프로그램 및 상기 사용자 응용 프로그램과 연관된 데이터를 검사하는 단계, 및 상기 검사 결과 상기 사용자 응용 프로그램 및 상기 데이터가 변조된 경우 이를 소정의 서버-상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결됨-에 보고하는 단계를 포함하는 사용자 응용 프로그램 제어 방법이 제공된다.

<25>       본 발명의 또 다른 일 측면에 따르면, 상기 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하는 단계, 소정의 시간 간격으로 소정의 서버에 접속하여 소정의 인증을 수행하는 단계, 및 상기 인증이 수행되지 않은 경우, 상기 파일 판독 요청을 처리하지 않음으로써 상기 사용자 응용 프로그램이 수행되지 않도록 하는 단계를 포함하는 사용자 응용 프로그램 제어 방법이 제공된다.

- <26> 본 발명의 또 다른 일 측면에 따르면, 상기 사용자 응용 프로그램이 접근하는 데이터 파일 내 복수 개의 데이터 블록 중 적어도 하나를 소정의 서버에 우선 파일(priority file)로 저장하는 단계, 상기 복수 개의 데이터 블록에 대한 오프셋, 데이터 블록의 크기 및 상기 데이터 블록에 연관되는 우선 파일 식별자를 인덱스 저장 수단에 저장하는 단계, 상기 사용자 응용 프로그램으로부터의 상기 데이터 파일 중 임의의 데이터 블록에 대한 판독 요구를 수신하는 단계, 인덱스 저장 수단을 참조하여 상기 임의의 데이터 블록에 대응하는 우선 파일을 식별하는 단계 및 상기 소정의 서버로부터 상기 우선 파일을 수신하여 상기 사용자 응용 프로그램에 전달하는 단계를 포함하는 사용자 응용 프로그램 제어 방법이 제공된다.
- <27> 이하에서는 첨부된 도면을 참조하여 본 발명의 실시예를 상세히 설명한다.
- <28> 도 1은 본 발명에 따라 클라이언트 컴퓨터와 파일 서버의 네트워크 연결을 도시한 도면이다.
- <29> 사용자들은 온라인 게임을 수행하기 위하여 자신의 클라이언트 컴퓨터(102, 103)에서 인터넷을 통하여 소정의 온라인 게임 서버(104)에 접속한다. 종래의 패키지 형태의 게임 소프트웨어인 경우에는 본 발명에 따라 간단히 온라인화 되어 게임의 수행 중 게임 서버(104)에 접속하여 게임을 수행하게 된다. 즉, 종래의 패키지 게임의 경우에도 사용자가 CD 등을 구입하지 않은 채 단지 게임 서버(104)에만 접속하여 본 발명에 따라 게임의 전체 파일 및 데이터가 아닌 일부의 파일 또는 데이터만을 다운로드 받은 후 바로 게임을 시작할 수 있다.
- <30> 도 2는 본 발명에 따라 클라이언트 컴퓨터에 포함된 사용자 응용 프로그램, 온라인 스트리밍 파일 라이브러리 및 네트워크 인터페이스의 관계를 도시한 블록도이다.

<31> 사용자 응용 프로그램(user application program)(201)은 종래의 패키지 소프트웨어 그대로이거나, 또는 본 발명에 따라 종래의 패키지 소프트웨어가 일부 수정된 프로그램이다. 또는 사용자 응용 프로그램(201)은 종래의 온라인 소프트웨어일 수도 있다. 본 발명은 종래의 패키지 소프트웨어를 온라인화하여 여러 가지 유용한 효과를 얻기 위한 것이나, 본 발명의 일부 사상은 종래의 온라인 소프트웨어에도 용이하게 적용되어 유용한 효과를 얻을 수 있다.

<32> 온라인 스트리밍 파일 라이브러리(online streaming file library)(202)는 사용자 응용 프로그램(201)의 파일 판독 요청 및 파일 기록 요청을 후킹(hooking)하고, 이러한 파일 판독 요청 및 파일 기록 요청을 수행하며, 또한 서버(104)와의 연결 유지, 불법적인 소프트웨어 사용의 모니터링 등의 기능을 수행한다. 또한, 온라인 스트리밍 파일 라이브러리는 필요한 데이터를 서버 또는 네트워크 상 인접한 다른 클라이언트 컴퓨터로부터 다운로드하는 기능을 수행한다. 온라인 스트리밍 파일 라이브러리는 라이브러리 이외의 다른 형태로 구현될 수도 있다.

<33> 온라인 스트리밍 파일 라이브러리(202)는 기존의 운영 체제 수준의 파일 입출력 API를 후킹한다. 기존 운영 체제 수준의 파일 입출력 API의 예로는 마이크로소프트사의 윈도우 운영 체제에 있어서의 CreateFile(), ReadFile(), WriteFile() 등의 함수가 있다. 윈도우 운영 체제에서는 모든 Win32 API 함수들에 대하여 임포트 섹션(Import Section)을 가진다. 임포트 섹션은 일종의 점프 벡터 테이블(Jump Vector Table)이다. 예를 들어, CreateFile() 함수의 경우, 임포트 섹션에 상기 CreateFile() 함수에 대한 포인터가 저장된다. 따라서 실행 파일(윈도우 운영 체제에서는 .exe의 확장자를 가지는 파일)에서 CreateFile() 함수를 사용하는 경우, 상기 실행 파일은 상기 임포트 섹션을

참조하여 CreateFile() 함수의 포인터를 얻게 되고, 이 포인터가 가리키는 함수를 수행하게 된다. API 후킹(API Hooking)이란 상기 임포트 섹션에서의 각 입출력 함수의 주소를 다른 함수의 주소로 변경함으로써 수행된다. 예를 들어, 임포트 섹션에서 CreateFile() 함수의 포인터를 운영 체제가 제공하는 CreateFile() 함수의 주소를 가리키는 것이 아닌, 다른 함수의 주소를 가리키도록 변경하면, 사용자 응용 프로그램에서 CreateFile()을 수행하는 경우 운영 체제가 제공하는 CreateFile() 함수가 아닌 상기 다른 함수를 수행하게 되는 것이다.

<34> 이러한 API 후킹을 수행하기 위하여 사용자 응용 프로그램의 메인 함수(main function)(예를 들어, 윈도우즈 시스템에서는 winmain() 함수)의 시작과 끝에 API 후킹 초기화 함수 및 API 후킹 종료 함수를 추가하여 API 후킹을 수행한다. API 후킹 초기화 함수는 본 발명에 의하여 제공되는 함수로써, 상기 사용자 응용 프로그램의 임포트 섹션에서의 각 입출력 API 함수의 포인터를 본 발명에 따른 입출력 함수의 포인터로 변경하는 기능을 수행한다. API 후킹 종료 함수에서는 상기 변경된 임포트 섹션을 원상으로 복구하는 기능을 수행한다. 따라서 본 발명에 따르면 종래의 패키지 소프트웨어의 메인 함수에 2가지 함수만을 추가함으로써 패키지 소프트웨어의 온라인화가 이루어진다. 본 발명의 또 다른 실시예에 따르면, 메인 함수의 엔트리 포인트(entry point) 자체를 후킹한다. 이 방법은 한 예로는 컴퓨터 바이러스 프로그램이 자신의 코드를 실행 파일에 추가하는 것과 유사한 방법이다. 본 실시예에 따르면, 응용 프로그램을 전혀 수정하지 않고서도 패키지 소프트웨어의 온라인화가 가능한 장점이 있다.

<35> 도 3은 본 발명에 따라 클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램으로부터의 파일 판독 요청을 처리하는 순서를 도시한 순서도이다.

<36> 먼저 본 발명에 따라 API 후킹을 위한 입출력 API 함수의 점프 벡터 테이블이 본 발명의 입출력 API 함수를 수행하도록 변경되어 있다. 단계(301)에서 사용자 응용 프로그램(201)으로부터 파일 판독 요청을 수신하고, 이를 후킹(hooking)한다. 즉, 사용자 응용 프로그램(201) 중 ReadFile() 함수를 수행하게 되는데, 상기 ReadFile() 함수의 수행이 후킹되어 본 발명에 따른 온라인 스트리밍 파일 입출력 API 함수(online streaming File I/O API function)가 수행된다. 본 발명에 따른 온라인 스트리밍 파일 입출력 API 함수들은 온라인 스트리밍 파일 라이브러리(202)에 저장된다.

<37> 본 발명에 따른 온라인 스트리밍 파일 입출력 API 함수는, 단계(302)에서, 상기 사용자 응용 프로그램(201)이 요구한 파일이 상기 사용자 응용 프로그램(201)이 저장되어 있는 클라이언트 컴퓨터에 저장되어 있는지 판단한다. 만일 상기 사용자 응용 프로그램(201)이 요구한 파일이 상기 클라이언트 컴퓨터에 저장되어 있으면, 단계(303)에서 상기 파일의 데이터를 판독하여 상기 파일의 데이터를 사용자 응용 프로그램(201)에 전달한다. 상기 파일의 데이터는 파일 전체의 데이터일 수도 있고, 상기 파일에 포함된 데이터 중 일부의 데이터일 수도 있다. 예를 들어 사용자 응용 프로그램이 소정의 게임 프로그램인 경우 상기 게임 프로그램이 수행되기 위하여 필요한 그래픽 데이터를 저장하고 있는 파일이 있고, 상기 게임 프로그램이 상기 그래픽 데이터를 필요로 하여 판독 요청을 하였는데, 상기 그래픽 데이터를 저장하고 있는 파일이 클라이언트 컴퓨터에 저장되어 있으면 이를 판독하여 사용자 응용 프로그램에 전달하는 것이다.

<38> 단계(302)에서 로컬 클라이언트 컴퓨터에 사용자 응용 프로그램(201)이 요구한 데이터가 저장되어 있지 않으면, 온라인 스트리밍 파일 입출력 API 함수는 단계(304)에서

네트워크 인터페이스(203)를 통하여 소정의 서버(104)에 접속한다. 상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결된다. 상기 소정의 서버는 상기 사용자 응용 프로그램(201)이 필요로 하는 모든 데이터를 저장하고 있다.

<39> 단계(305)에서 온라인 스트리밍 파일 입출력 API 함수는 서버(104)로부터 파일의 데이터를 수신하고, 단계(306)에서 상기 수신된 데이터를 로컬 클라이언트 컴퓨터에 캐싱한다. 이러한 캐싱을 통하여 동일한 데이터를 중복하여 다운로드 하지 않을 뿐더러 후술하는 우선 파일의 개념을 이용하여 앞으로 다운로드 할 것으로 예상되는 데이터를 미리 다운로드 해 두는 이점이 있다. 예를 들어 현재 필요한 데이터는 파일 1의 블록 5인 경우, 이전에 사용자 응용 프로그램(201)을 수행하면서 그 동작을 조사하여 본 결과 블록 5와 블록 6을 함께 사용자 응용 프로그램(201)이 사용한다면, 블록 5와 블록 6을 함께 서버(104)로부터 다운로드 받아 이를 캐싱 해 두는 것이다.

<40> 단계(307)에서 온라인 스트리밍 파일 입출력 API 함수는 서버(104)로부터 수신한 파일의 데이터를 사용자 응용 프로그램(201)에 전달한다. 그러면 사용자 응용 프로그램(201)은 계속해서 수행되게 된다.

<41> 예를 들어, 게임 소프트웨어의 전체 수행에 필요한 데이터 파일들의 전체 크기가 200 Mbyte이지만, 게임 소프트웨어가 현재의 장면을 디스플레이하기 위하여 필요한 그래픽 데이터는 3Mbyte이고, 이 그래픽 데이터가 로컬 클라이언트 컴퓨터에 저장되어 있지 않다면, 이를 저장하고 있는 서버에 접속하여 상기 필요한 그래픽 데이터를 다운로드 하여 이를 사용하는 것이다. 따라서, 본 발명에 따르면 사용자는 초기에 대용량의 데이터 파일을 다운로드 하지 않고서도 쉽게 온라인 게임을 즐길 수 있다. 또한 패키지 게임 소프트웨어를 본 발명에 따라 온라인화한 경우에도, 필요한 데이터의 일부만을 그때 그

때 서버로부터 다운로드 하여 게임을 수행할 수 있다. 즉, 본 발명에 따르면, 사용자 응용 프로그램의 수행에 필요한 전체 데이터를 미리 로컬 클라이언트 컴퓨터에 저장하여 두지 않고서도 사용자 응용 프로그램을 수행할 수 있다.

<42> 또한 본 발명의 또 다른 실시예에 따르면, 파일의 데이터의 판독 및 수신을 파일의 데이터 중 일부의 데이터를 수신하여 수행하고, 이러한 일부의 데이터의 수신은 상기 파일의 오프셋 및 상기 일부의 데이터의 크기를 이용하여 수행된다. 이에 관하여는 도 5 및 도 6을 참조하여 뒤에 자세히 설명한다.

<43> 본 발명의 또 다른 실시예에 따르면, 하나의 클라이언트 컴퓨터(102) 상에서 수행되고 있는 사용자 응용 프로그램(102)이 파일 판독 요청을 하고 있지 않는 동안에 네트워크 상으로 인접한 위치에 위치한 다른 클라이언트 컴퓨터(103)에 사용자 응용 프로그램(102)이 수행하기에 필요한 데이터가 저장되어 있는지 식별하고, 만일 다른 클라이언트 컴퓨터(103)에 필요한 데이터가 저장되어 있으면 이를 수신하여 클라이언트 컴퓨터(102)에 저장하여 둔다. 다른 클라이언트 컴퓨터(103)도 마찬가지로 클라이언트 컴퓨터(102)로부터 데이터를 수신할 수도 있다. 이의 수행을 위한 프로그램은 사용자 응용 프로그램(102)이 시작되면 동시에 백그라운드로 시작될 수도 있고, 또는 사용자 응용 프로그램(102)의 시작 및 종료와 무관하게 클라이언트 컴퓨터(102, 103)에서 시작되어 수행될 수도 있다. 본 발명에 따르면, 사용자 응용 프로그램이 데이터를 요구하기 전에 미리 필요한 데이터를 다운로드 받아 로컬 클라이언트 컴퓨터에 저장하여 두기 때문에, 사용자 응용 프로그램의 수행을 효율적으로 할 수 있을 뿐만 아니라 서버(104)에 대한 부하도 경감시킨다.



<44> 본 발명의 또 다른 실시예에 따르면, 사용자 응용 프로그램(102)으로부터 데이터 판독 요청을 후킹하고, 상기 데이터가 로컬 클라이언트 컴퓨터에 저장되어 있으면 이를 사용자 응용 프로그램(102)에 전달하고, 로컬 클라이언트 컴퓨터에 저장되어 있지 않으면 네트워크 상 인접한 다른 클라이언트 컴퓨터에 저장되어 있는지 판단한다. 만일 다른 클라이언트 컴퓨터에 저장되어 있으면 P2P를 이용하여 이를 다운로드 한다. 만일 클라이언트 컴퓨터에 저장되어 있지 않으면 서버에 접속하여 다운로드 한다.

<45> 본 발명의 또 다른 실시예에 따르면, 사용자 응용 프로그램(102)으로부터의 파일 판독 요청을 후킹하고, 상기 파일 판독 요청에 응답하여 클라이언트 컴퓨터에 저장된 사용자 응용 프로그램(102) 및 사용자 응용 프로그램(102)과 연관된 데이터를 검사한다. 만약 상기 사용자 응용 프로그램(102) 또는 상기 데이터가 변조된 경우에는 상기 사용자 응용 프로그램(102) 또는 데이터가 해킹 또는 크래킹된 것으로 볼 수 있으므로 이를 서버(104)에 보고한다. 상기 변조 여부의 판단은 CRC 체크 등 데이터의 변조 여부를 체크하는 다양한 방법이 이용될 수 있다. 따라서 본 발명에 따르면, 사용자 응용 프로그램의 해킹 또는 크래킹 여부를 확인하여 이에 대한 적절한 대응을 할 수 있다. 본 발명에 따른 사용자 응용 프로그램 및 데이터의 검사는 일정한 시간 간격으로 수행될 수도 있다.

<46> 본 발명의 또 다른 실시예에 따르면, 사용자 응용 프로그램(102)으로부터의 파일 판독 요청을 후킹하고, 소정의 시간 간격으로 서버(104)에 접속하여 소정의 인증을 수행한다. 상기 소정의 시간 간격은 일정한 시간 간격일 수도 있고 일정하지 않은 시간 간격일 수도 있다. 상기 인증은 상기 사용자 응용 프로그램의 사용자가 적법한 사용자인지에 관한 인증이다. 만일 상기 인증이 실패하면 그 이후의 사용자 응용 프로그램의 파

일 판독 요청에 대하여 온라인 스트리밍 파일 입출력 API 함수는 아무런 처리를 수행하지 않거나 에러를 사용자 디스플레이에 표시하여 사용자 응용 프로그램이 적절히 수행되지 못하도록 한다. 본 실시예에 따르면, 사용자의 불법 복제 등에 의한 불법적인 프로그램의 사용을 막을 수 있다. 또한, 이러한 본 발명에 따르면 패키지 프로그램을 쉽게 온라인화하여 사용 시간에 비례한 과금을 수행할 수 있다. 즉, 패키지 프로그램은 사용 시간에 따른 과금이 어려운데, 이를 본 발명에 따라 온라인화하고, 사용자 응용 프로그램의 사용 시간을 서버(104)에서 관리함으로써 사용자에게 대하여 사용 시간에 따른 과금을 수행할 수 있다. 보통 패키지 게임의 경우 1달 정도면 게임을 모두 끝내고 더 이상 게임을 하지 않는 경우가 많기 때문에, 상기 1달 정도의 사용 요금만 사용자가 지불하면 상기 게임을 즐길 수 있다. 따라서 사용자는 저렴한 비용으로 게임 프로그램을 즐길 수 있는 장점이 있다.

<47> 도 4는 본 발명에 따라 클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램으로부터의 파일 기록 요청을 처리하는 순서를 도시한 순서도이다.

<48> 단계(401)에서 사용자 응용 프로그램(201)으로부터 파일 기록 요청을 수신하고, 이를 후킹(hooking)한다. 즉, 사용자 응용 프로그램(201) 중 WriteFile() 함수를 수행하게 되는데, 상기 WriteFile() 함수의 수행이 후킹되어 본 발명에 따른 온라인 스트리밍 파일 입출력 API 함수가 수행된다. 상기 WriteFile() 함수에 대응하는 온라인 스트리밍 파일 입출력 API 함수는 온라인 스트리밍 파일 라이브러리(202)에 저장된다.

<49> 본 발명에 따른 온라인 스트리밍 파일 입출력 API 함수는, 단계(402)에서, 상기 사용자 응용 프로그램(201)이 기록 요청한 파일이 서버(104)에 업로드(upload)할 필요가 있는지 판단한다. 만일 상기 사용자 응용 프로그램(201)이 기록 요청한 파일이 서버

(104)에 업로드 할 필요가 없으면, 단계(404)에서 상기 사용자 응용 프로그램(201)은 상기 기록 요청한 파일을 로컬 클라이언트 컴퓨터에 저장한다. 예를 들어, 사용자에게 의존적인 데이터 중 서버(104)에서 관리할 필요가 없는 데이터는 로컬 클라이언트 컴퓨터에만 저장되어 관리될 수 있다. 이와 같이 어떤 데이터가 서버에 업로드 될 필요가 있는지의 여부는 본 발명에 따른 컴퓨터 프로그램의 제공자가 결정하여 그에 따른 로직을 상기 온라인 스트리밍 파일 입출력 API 함수에 포함한다.

<50> 만일 상기 사용자 응용 프로그램(201)이 기록 요청한 파일이 서버(104)에 업로드 할 필요가 없으면, 온라인 스트리밍 파일 입출력 API 함수 단계(403)에서 네트워크 인터페이스(203)를 통하여 소정의 서버(104)에 접속한다. 상기 소정의 서버(104)는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결된다.

<51> 단계(405)에서 온라인 스트리밍 파일 입출력 API 함수는 서버(104)에 업로드 할 파일의 데이터를 로컬 클라이언트 컴퓨터에 캐싱한다. 이러한 캐싱을 통하여 동일한 데이터를 판독하게 되는 경우 다시 서버(104)로부터 다운로드 하지 않아도 되기 때문에, 프로그램 수행의 효율성을 높인다.

<52> 단계(406)에서 온라인 스트리밍 파일 입출력 API 함수는 업로드 할 파일의 데이터를 서버(104)에 업로드 한다.

<53> 도 5는 본 발명에 따라 사용자 응용 프로그램이 접근하는 파일 내 데이터 블록을 프로파일링(profiling)하여 우선 파일(priority file)을 구성하는 경우, 데이터 블록과 우선 파일의 관계를 도시한 도면이다.

- <54>        본 실시예의 설명을 용이하게 하기 위하여, 사용자 응용 프로그램은 게임을 수행하는 프로그램으로 프로그램명이 game.exe라고 가정하고, 상기 game.exe가 필요로 하는 데이터 파일들 중 하나를 data1.dat이라고 가정한다. 상기 data1.dat은 10Mbyte 크기의 데이터 파일이다. 상기 game.exe의 메인 함수는 일부 수정되어 파일 입출력 API 함수는 온라인 스트리밍 파일 입출력 API 함수에 의하여 후킹된다.
- <55>        먼저 사용자 응용 프로그램이 순차적으로 참조하는 데이터를 찾기 위하여 프로파일링(profiling)하는 단계를 설명한다.
- <56>        사용자 응용 프로그램(201)을 수행시키면서, 상기 사용자 응용 프로그램(201)이 참조하는 데이터를 식별한다. 본 발명의 일 실시예에 따르면, 상기 사용자 응용 프로그램(201)이 수행되면서 일정한 시간 동안 참조되는 데이터를 묶어서 하나의 파일로 저장하는데, 이는 본 발명에서의 우선 파일(priority file)의 하나의 예이다. 우선 파일은 하나의 파일 내에 저장된 일부의 데이터 또는 복수의 파일 내에 저장된 일부의 데이터의 조합일 수 있다. 즉, 파일 1에 포함된 데이터 블록 A, 파일 2에 포함된 데이터 블록 B를 조합하여 우선 파일 하나를 생성할 수 있다. 이렇게 되면, 사용자 응용 프로그램을 수행하면서 다운로드 되는 데이터의 구조를 파악하기 어렵기 때문에 해킹 및 크래킹이 더욱 어려워지는 장점이 있다.
- <57>        Game.exe를 수행하면서 5초 단위로 참조된 데이터를 묶어서 하나의 우선 파일을 생성한다. 예를 들어, game.exe를 수행하여 0초부터 5초 내에 판독 요청이 발생한 데이터가 data1.dat 파일의 오프셋 100부터 1024 바이트 크기의 데이터이면, 상기 data1.dat 파일의 오프셋 100부터 1024 바이트 크기의 데이터를 우선 파일 0으로 저장한다. 다음 game.exe가 수행되어 5초부터 10초 사이에 판독 요청이 발생한 데이터는 우선 파일

1(502)에 저장된다. 도 5의 실시예는 game.exe가 수행되어 5초부터 15초 사이에 발생한 판독 요청이 없는 경우이다. Game.exe가 수행되어 15초부터 20초 사이에 판독 요청이 발생한 데이터가 data1.dat 파일의 오프셋 2000에서부터 4096 바이트 크기의 데이터인데, 이는 우선 파일 3(503)으로 저장된다. 본 실시예에서는 5초 단위로 판독 요구가 발생한 데이터를 하나씩의 파일로 저장하고, 판독 요구가 없는 경우에는 우선 파일을 생성하지 않고, 우선 파일의 번호만 증가시켰지만, 본 발명의 또 다른 실시예에 따르면, 상기 시간 단위를 다르게 하거나, 우선 파일의 생성 방법을 다르게 하는 것도 가능하다. 예를 들어, 우선 파일 0의 생성 후, 그 다음에 생성되는 우선 파일은 우선 파일 1, 2 등의 순으로 하여 중간에 비어 있는 우선 파일이 없도록 할 수도 있다. 어느 경우이든지 사용자 응용 프로그램이 수행되면서 먼저 요구될 가능성이 높은 데이터를 우선 순위가 높은 데이터로 지정하여 사용한다. 즉, 우선 순위 별로 별도의 파일로 관리할 수도 있고, 하나의 파일 내에서 각 데이터 블록 별로 우선 순위만을 주고 이 우선 순위만을 관리하여 필요한 데이터 블록을 판독하여 사용자 응용 프로그램에 제공할 수도 있다. 이렇게 생성된 우선 파일은 서버(104)에 저장된다.

<58> 본 발명의 또 다른 실시예에 따르면, 프로그램이 수행되면서 먼저 요구될 가능성이 높은 데이터 블록의 순으로 우선 순위를 주지 않고, 데이터의 크기 등 다른 요소를 고려하여 우선 순위를 정할 수 있다. 사용자 응용 프로그램으로부터의 데이터 판독 요구가 있는 경우 우선 순위가 다른 데이터가 요구된 경우에는 우선 순위가 높은 데이터의 순으로 데이터 판독 요구를 처리하고, 동일한 우선 순위에 대해서는 먼저 요구된 데이터 판독 요구를 수행한다.

- <59> 또한 본 발명의 또 다른 실시예에 따르면, 이러한 우선 순위에 따라 우선 순위가 높은 경우에는 사용자 응용 프로그램으로부터의 데이터 판독 요구를 수신하는 즉시 해당 데이터를 서버(104)로부터 수신하지만, 우선 순위가 낮은 경우에는 인접한 다른 클라이언트 컴퓨터로부터 백그라운드 P2P를 이용하여 다운로드 한다.
- <60> 이와 같이 우선 파일을 생성하여 두면, 사용자 응용 프로그램이 시작되면서 필요한 데이터의 순으로 우선 파일이 생성되게 된다. 따라서, 통계적으로 사용자 응용 프로그램이 우선 파일 0 다음에는 우선 파일 3의 데이터를 요구할 것이 예상되므로, 사용자 응용 프로그램이 판독 요구를 하고 있지 않은 동안, 예를 들어 상기 실시예에서 game.exe 시작 후 5초에서 15초 사이에, 우선 파일 3을 백그라운드로 미리 로컬 클라이언트 컴퓨터에 다운로드 해 두면, 사용자가 사용자 응용 프로그램을 수행할 때 필요한 데이터가 미리 로컬 클라이언트 컴퓨터에 다운로드 되어 있기 때문에, 필요 시에 다운로드를 기다릴 필요가 없다.
- <61> 또한 각 사용자에게 따라 사용자 응용 프로그램을 수행하면서 서로 다른 명령을 먼저 수행할 수 있기 때문에, 각 사용자 응용 프로그램이 요청하는 데이터의 순서가 서로 다를 수 있다. 본 발명의 또 다른 실시예에 따르면, 이러한 경우를 대비하여 다수의 사용자의 사용자 응용 프로그램 사용 패턴을 분석하고 이를 반영하여 우선 파일을 생성한다. 이를 위하여 종래 기술에 따른 다양한 통계학적 방법이 이용될 수 있다. 예를 들어 다수의 사용자로 하여금 사용자 응용 프로그램을 수행하게 하고, 각 사용자에게 따른 요청 데이터를 분석하여 먼저 요구되는 데이터일수록 일정한 가중치를 주어, 각 시간대 별로 가장 많이 요청된 데이터를 그 시간 대의 우선 파일로 하여 저장할 수 있다. 본 발명의 또 다른 실시예에 따르면, 최초에는 우선 임의로 우선 파일을 생성한 후, 사용자들이 사

용자 응용 프로그램을 사용하면서 서버(104)에 요청하는 데이터의 빈도를 보고, 가장 많이 요구되는 데이터의 순으로 우선 순위를 주어 우선 파일을 생성해 둔다.

<62> 도 6은 본 발명의 일 실시예에 따라 구성된 인덱스 테이블을 도시한 도면이다.

<63> 인덱스 테이블(index table)(600)은 우선 파일이 어느 오프셋에서부터 어느 크기의 데이터 블록에 대응하는지에 관한 정보를 포함하며, 상기 프로파일링 시에 생성된다. 도 6의 인덱스 테이블은 데이터 파일이 하나인 것을 가정하고, 데이터 파일에 관한 정보가 포함되어 있지 않으나, 데이터 파일이 복수 개인 경우에는 우선 파일에 대응하는 데이터 파일이 무엇인지에 관한 정보도 포함한다. 도 6은 도 5의 프로파일링에 따라 구성된 우선 파일에 대한 인덱스 테이블(600)이다.

<64> 이제 도 6의 인덱스 테이블(600) 및 도 3을 참조하여 사용자 응용 프로그램이 데이터 판독 요청을 한 경우의 처리에 관하여 상세히 설명한다.

<65> 사용자 응용 프로그램인 game.exe가 데이터 파일인 data1.dat의 오프셋 2500부터 1024 바이트의 데이터를 요구하는 파일 판독 요청을 한 경우를 예로 설명한다. 사용자 응용 프로그램은 ReadFile() 함수를 사용하여 특정 파일의 특정 오프셋부터 일정한 크기의 데이터의 판독을 요청할 수 있다. 이러한 사용자 응용 프로그램의 ReadFile() 함수는 온라인 스트리밍 파일 입출력 API 함수에 의하여 후킹된다.

<66> 온라인 스트리밍 파일 입출력 API 함수는 상기 요구되는 데이터가 어느 우선 파일에 저장되어 있는지를 판단하기 위하여 인덱스 테이블(600)을 참조한다. 인덱스 테이블(600)을 참조한 결과, 상기 데이터에 대응하는 우선 파일은 우선 파일 3임이 식별

된다. 온라인 스트리밍 파일 입출력 API 함수는 상기 우선 파일 3이 로컬 클라이언트 컴퓨터에 저장되어 있는지 확인한다.

<67> 만일 상기 우선 파일 3이 상기 로컬 클라이언트 컴퓨터에 저장되어 있으면, 상기 파일의 데이터를 판독하여 game.exe가 요청한 1024 바이트 크기의 데이터만을 game.exe에 전달한다.

<68> 만일 우선 파일 3이 로컬 클라이언트 컴퓨터에 저장되어 있지 않으면, 온라인 스트리밍 파일 입출력 API 함수는 네트워크 인터페이스(203)를 통하여 소정의 서버(104)에 접속한다. 상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결되고 모든 우선 파일을 저장하고 있다. 본 발명의 일 실시예에 따르면 상기 서버(104)는 우선 파일에 포함되지 않은 데이터들도 모두 저장한다.

<69> 온라인 스트리밍 파일 입출력 API 함수는 서버(104)로부터 우선 파일 3을 수신하고, 상기 수신된 우선 파일 3을 로컬 클라이언트 컴퓨터에 캐싱한다. 이러한 캐싱을 통하여 사용자 응용 프로그램이 우선 파일 3에 포함된 다른 데이터를 요청하는 경우에 다시 우선 파일 3을 다운로드 할 필요가 없게 된다.

<70> 온라인 스트리밍 파일 입출력 API 함수는 서버(104)로부터 수신한 우선 파일 3을 판독하여 그 중 game.exe가 요구한 데이터를 사용자 응용 프로그램(201)에 전달한다. 그러면 사용자 응용 프로그램(201)은 계속해서 수행되게 된다.

<71> 이상의 실시예들에서 사용자 응용 프로그램으로부터 파일에 대한 판독 요청을 수신하는 구성은 사용자 응용 프로그램으로부터의 파일 판독 요청을 후킹(hooking)하는 방법으로 설명되었다. 그러나, 본 기재에 의하여, 본 발명이 속하는 기술분야에서 통상의



지식을 가진 자는 후킹뿐 아니라 파일 입출력 함수를 다른 함수로 매핑하는 함수 리매핑(function re-mapping) 방법을 사용하여 본 발명을 구현할 수도 있으며, 기타 균등 범위 내의 다른 방법들을 채용할 수도 있다.

<72> 또한 본 발명의 실시예들은 다양한 컴퓨터로 구현되는 동작을 수행하기 위한 프로그램 명령을 포함하는 컴퓨터 판독 가능 매체를 포함한다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 상기 매체는 프로그램 명령, 데이터 구조 등을 지정하는 신호를 전송하는 반송파를 포함하는 광 또는 금속선, 도파관 등의 전송 매체일 수도 있다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

<73> 이상과 같이 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 이는 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다. 따라서, 본 발

명 사상은 아래에 기재된 특허청구범위에 의해서만 파악되어야 하고, 이의 균등 또는 등가적 변형 모두는 본 발명 사상의 범주에 속한다고 할 것이다.

### 【발명의 효과】

<74> 본 발명에 따르면, 종래의 패키지 형태의 소프트웨어 제품을 간편하게 온라인 소프트웨어화하여 온라인 스트리밍(on-line streaming)의 효과를 가져오고, 네트워크를 통하여 서버와의 연결을 유지함으로써 외부로부터의 해킹 및 크래킹을 효과적으로 방지하여 불법적인 소프트웨어의 사용을 방지할 수 있다.

<75> 또한, 본 발명은 종래의 패키지 소프트웨어를 간단히 온라인화함으로써, 주기적으로 상기 패키지 소프트웨어의 불법 복제 여부를 감시할 수 있고, 만일 불법 복제 제품인 경우에는 상기 패키지 소프트웨어의 동작을 정지시킴으로써, 불법 복제를 막을 수 있다. 또한 본 발명에 따르면, 패키지 소프트웨어를 온라인화함으로써 패키지 소프트웨어의 과금을 사용 시간 당 과금이 가능하도록 한다.

<76> 또한 본 발명에 따르면, 온라인 소프트웨어에서 파일 전체가 아닌 파일의 블록 단위로 다운로드 하여 초기에 다운로드 하여야 하는 데이터의 크기를 최소화시킴으로써, 사용자들이 초기에 쉽게 게임을 즐길 수 있도록 한다. 즉, 본 발명에 따르면 프로그램의 시작 전 모든 프로그램이 로컬 컴퓨터에 저장되어 있지 않아도 수행될 수 있으며, 요청시 데이터 요구(on demand data request)를 가능하게 한다.

<77> 또한 본 발명에 따르면, 온라인 소프트웨어가 요구하는 데이터의 패턴을 조사하여, 이에 따라 다운로드 되는 데이터를 최적화하고, 사용자 응용 프로그램이 데이터의 판독을 요청하지 않는 경우에도, 상기 조사된 데이터의 패턴에 따라 미리 백그라운드

피투피(Background P2P)를 통하여 필요한 데이터를 미리 다운로드 받음으로써 사용자가 데이터의 다운로드를 위하여 기다리는 시간을 최소화하고, 또한 서버의 대역폭(bandwidth) 사용을 최소화한다.

**【특허청구범위】****【청구항 1】**

클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램을 제어하기 위한 방법에 있어서,

사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하는 단계;

상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있는지 판단하는 단계;

상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있으면, 상기 파일의 데이터를 상기 사용자 응용 프로그램에게 전달하는 단계; 및

상기 파일이 상기 클라이언트 컴퓨터에 저장되어 있지 않으면, 상기 파일이 저장되어 있는 소정의 서버-상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결됨-로부터 상기 파일의 데이터 중 일부의 데이터를 수신하여 상기 사용자 응용 프로그램에 전달하는 단계

를 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

**【청구항 2】**

제1항에 있어서,

상기 파일의 데이터 중 일부의 데이터의 수신은 상기 파일의 오프셋 및 상기 일부의 데이터의 크기를 이용하여 수행되는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

**【청구항 3】**

제1항에 있어서,

상기 일부의 데이터만 수신하고 상기 파일의 전체 데이터를 수신하지 않은 상태에서 상기 사용자 응용 프로그램을 계속 수행하는 단계

를 더 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

**【청구항 4】**

제1항에 있어서,

상기 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하는 단계는, 상기 사용자 응용 프로그램으로부터의 상기 파일에 대한 상기 파일 판독 요청을 후킹(hooking)하거나 상기 파일 판독 요청을 처리하는 원래의 함수를 다른 함수로 매핑하는 단계를 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

**【청구항 5】**

제1항에 있어서,

상기 파일이 저장되어 있는 소정의 서버로부터 상기 파일의 데이터를 수신하여 상기 클라이언트 컴퓨터에 캐싱하는 단계

를 더 포함하는 것을 특징으로 하는 파일 접근 제공 방법.

**【청구항 6】**

제1항에 있어서,

상기 파일을 저장하고 있는 제2 클라이언트 컴퓨터를 식별하는 단계; 및

상기 제2 클라이언트 컴퓨터로부터 상기 파일을 수신하여 상기 사용자 응용 프로그램에 전달하는 단계

를 더 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

**【청구항 7】**

제1항에 있어서,

상기 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하지 않는 경우, 상기 사용자 응용 프로그램이 필요로 할 것으로 예상되는 데이터를 상기 데이터를 저장하고 있는 제2 클라이언트 컴퓨터로부터 수신하여 저장하는 단계

를 더 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

**【청구항 8】**

제7항에 있어서,

상기 사용자 응용 프로그램이 필요로 할 것으로 예상되는 데이터의 판단은 이전에 상기 사용자 응용 프로그램을 실행한 경우 상기 사용자 응용 프로그램이 요구하였던 데이터에 기초하여 결정되는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

**【청구항 9】**

클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램을 제어하기 위한 방법에 있어서,

사용자 응용 프로그램으로부터의 파일에 대한 파일 기록 요청을 수신하는 단계;

상기 파일이 소정의 서버-상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결됨-에 업로드(upload)될 필요가 있는지 판단하는 단계;

상기 파일이 상기 소정의 서버에 업로드될 필요가 있으면, 상기 파일을 상기 소정의 서버로 업로드하는 단계; 및

상기 파일이 상기 소정의 서버에 업로드될 필요가 없으면, 상기 클라이언트 컴퓨터에 상기 파일을 기록하는 단계

를 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

#### 【청구항 10】

사용자 응용 프로그램을 제어하기 위한 방법에 있어서,

상기 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하는 단계;

상기 파일 판독 요청에 응답하여 클라이언트 컴퓨터에 저장된 상기 사용자 응용 프로그램 및 상기 사용자 응용 프로그램과 연관된 데이터를 검사하는 단계; 및

상기 검사 결과 상기 사용자 응용 프로그램 및 상기 데이터가 변조된 경우 이를 소정의 서버-상기 소정의 서버는 상기 클라이언트 컴퓨터와 네트워크를 통하여 연결됨-에 보고하는 단계

를 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

#### 【청구항 11】

제10항에 있어서,

상기 검사하는 단계는 CRC 체크를 이용하는 검사하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

#### 【청구항 12】

사용자 응용 프로그램을 제어하기 위한 방법에 있어서,  
 상기 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하는 단계;  
 소정의 시간 간격으로 소정의 서버에 접속하여 소정의 인증을 수행하는 단계; 및  
 상기 인증이 수행되지 않은 경우, 상기 파일 판독 요청을 처리하지 않음으로써 상기 사용자 응용 프로그램이 수행되지 않도록 하는 단계  
 를 포함하는 사용자 응용 프로그램 제어 방법.

#### 【청구항 13】

제12항에 있어서,  
 상기 사용자 응용 프로그램의 사용자에 대한 과금을 상기 사용자의 상기 사용자 응용 프로그램의 사용 시간에 기초하여 수행하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

#### 【청구항 14】

클라이언트 컴퓨터에서 수행되는 사용자 응용 프로그램을 제어하기 위한 방법에 있어서,  
 상기 사용자 응용 프로그램이 접근하는 데이터 파일 내 복수 개의 데이터 블록 중 적어도 하나를 소정의 서버에 우선 파일(priority file)로 저장하는 단계;



상기 복수 개의 데이터 블록에 대한 오프셋, 데이터 블록의 크기 및 상기 데이터 블록에 연관되는 우선 파일 식별자를 인덱스 저장 수단에 저장하는 단계;

상기 사용자 응용 프로그램으로부터의 상기 데이터 파일 중 임의의 데이터 블록에 대한 판독 요구를 수신하는 단계;

인덱스 저장 수단을 참조하여 상기 임의의 데이터 블록에 대응하는 우선 파일을 식별하는 단계; 및

상기 소정의 서버로부터 상기 우선 파일을 수신하여 상기 사용자 응용 프로그램에 전달하는 단계

를 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

#### 【청구항 15】

제14항에 있어서,

상기 우선 파일을 저장하고 있는 제2 클라이언트 컴퓨터를 식별하는 단계; 및

상기 제2 클라이언트 컴퓨터로부터 상기 우선 파일을 수신하는 단계

를 더 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

#### 【청구항 16】

제14항에 있어서,

상기 사용자 응용 프로그램으로부터의 파일에 대한 파일 판독 요청을 수신하지 않는 경우, 상기 사용자 응용 프로그램이 필요로 할 것으로 예상되는 우선 파일을 상기 우선 파일을 저장하고 있는 제2 클라이언트 컴퓨터로부터 수신하여 저장하는 단계

를 더 포함하는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

【청구항 17】

제16항에 있어서,

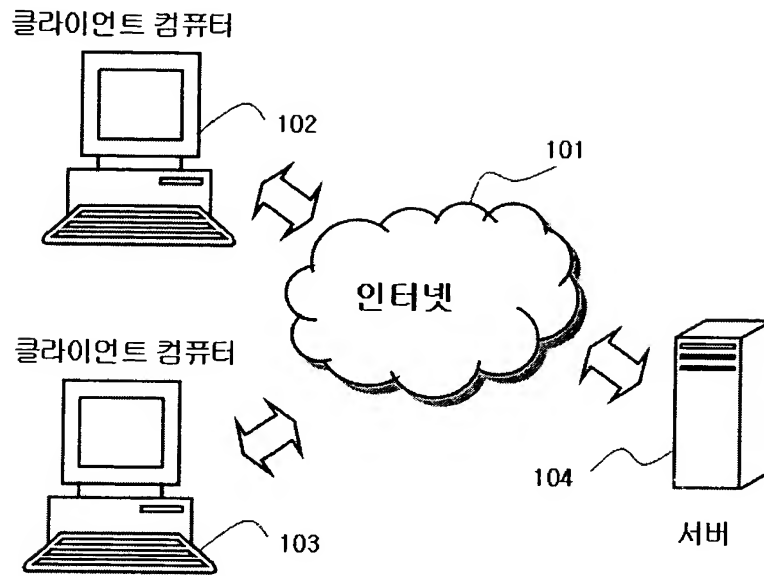
상기 사용자 응용 프로그램이 필요로 할 것으로 예상되는 우선 파일의 판단은 이전에 상기 사용자 응용 프로그램을 실행한 경우 상기 사용자 응용 프로그램이 요구하였던 우선 파일에 기초하여 결정되는 것을 특징으로 하는 사용자 응용 프로그램 제어 방법.

【청구항 18】

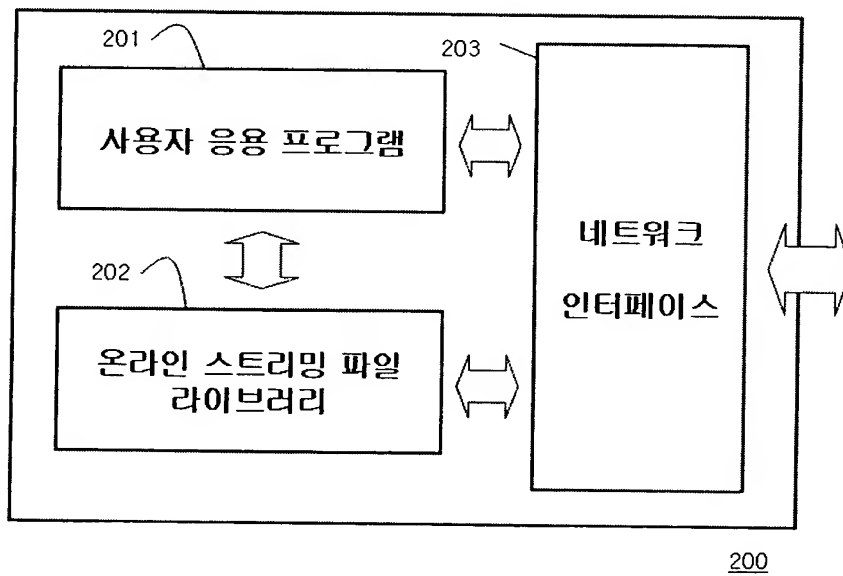
제1항 내지 제17항 중 어느 하나의 항의 방법을 실행시키기 위한 프로그램을 기록한 컴퓨터 판독 가능한 기록 매체.

## 【도면】

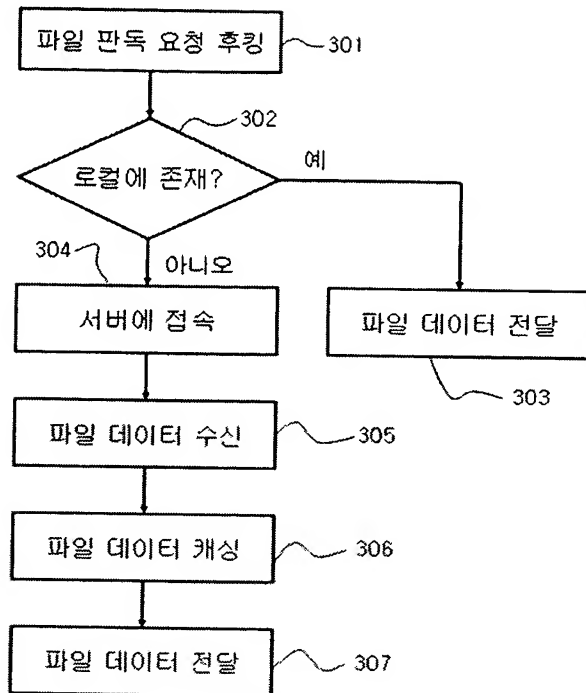
【도 1】



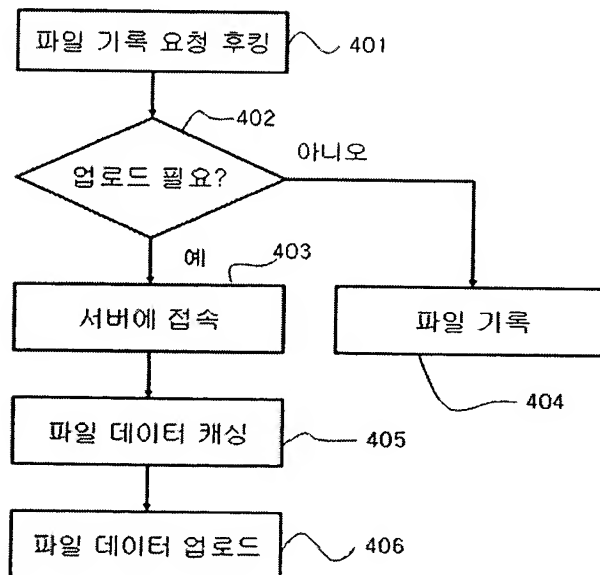
【도 2】



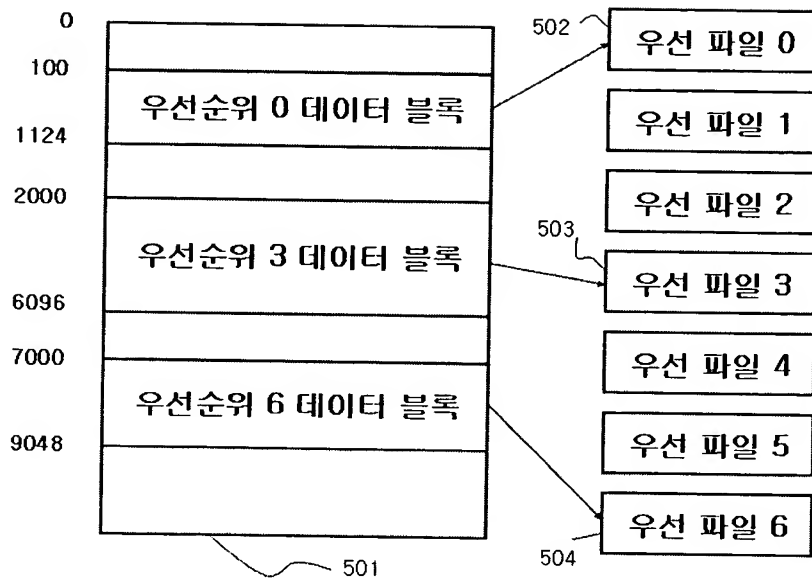
【도 3】



【도 4】



【도 5】



【도 6】

오프셋	데이터 블록 의 크기	우선 파일 식별자
100	1024	0
2000	4096	3
7000	2048	6

600